ISSN XXXX XXXX © 2018 IJESC

Research Article Volume 8 Issue No.4

Application of Deep Learning Techniques for Text Classification on Small Datasets

Anil Kumar Sharma[1], Souvik Hazra[2]
Assistant Professor[1], Ms Student[2]
Department of Mathematics[1], Department of Computer Science & Engineering[2]
Academy of Technology, India[1]
Eurecom, France[2]

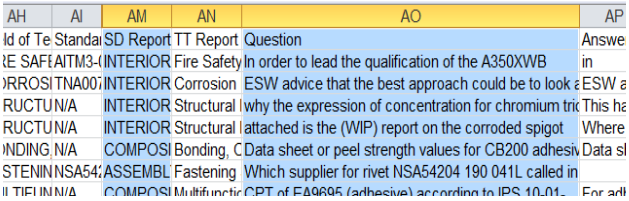
Abstract:
Text classification is a common NLP problem that involves building a model trained on supervised text and distribute labels to unseen text data. Deep learning have proven to give state of the art performance for large datasets of images and speech, motivated by that in this paper we study application of deep learning on a custom small text dataset by testing them using different architectures and embedding types.

Keywords: Text Categorization, Deep Learning, Text Classification

I. INTRODUCTION

Most NLP applications involve categorization of textual data at the foundation. Dictionaries, knowledge bases and special tree kernels are often used in Traditional text classifiers. In traditional supervised ML methods, a model is created based on a training set in which there are documents with tagged labels and the model is trained on this dataset and later used to predict labels for new documents. Depending on the classification algorithm or strategy used, the classifier might also provide a confidence measure to indicate how confident it is that the classification label is correct. In traditional text classification, a common methodology to do a pre-processing of text to make it suitable to be fed to classifier is TF-IDF (term frequency - inverse document frequency). The TF-IDF weighting for a word increases with the number of times the word appears in the document but decreases based on how frequently the word appears in the entire document set. Most text categorization algorithms represent a document collection as a Bag of Words (BOW). In the real world, several algorithms exist for classification such as Support Vector Machines (SVMs), Naive Bayes and Decision Trees. In recent years, deep learning techniques or deep networks are providing the state-of-art performance in large datasets of image analysis, speech recognition with the main advantage that the features needed for the task are automatically learned by the network; a prominent difference from previous methodologies. Motivated by this advantage and the good performance of deep networks in these domains, in this report, we investigate the use of deep learning technique in text classification for small datasets. To this end, we have used a custom questionnaire dataset in our investigation.

Our dataset contains a pool of question and labels highlighted below.



AH	AI	AM	AN	AO	AP
ɫld of Te	Standa	SD Report	TT Report	Question	Answe
ƦE SAFE	AITM3-(	INTERIOR	Fire Safety	In order to lead the qualification of the A350XWB	in
ƆRROSI	TNA007	INTERIOR	Corrosion	ESW advice that the best approach could be to look a	ESW a
RUCTU	N/A	INTERIOR	Structural I	why the expression of concentration for chromium tric	This ha
RUCTU	N/A	INTERIOR	Structural I	attached is the (WIP) report on the corroded spigot	Where
ƆNDING,	N/A	COMPOSI	Bonding, C	Data sheet or peel strength values for CB200 adhesiv	Data sl
STENIN	NSA542	ASSEMBL	Fastening	Which supplier for rivet NSA54204 190 041L called in	
ILTIFUN	N/A	COMPOSI	Multifunctic	CPT of FA9695 (adhesive) according to IPS 10-01-	For adh

Dataset Snapshot

We use it as a test case for exploring use of deep learning in text classification for small datasets, we try to map each of the questions into its specific SD REPORT class using different Deep learning Archetype. Our dataset contains 1800 records balanced among 3 categories.

II. USING LSTM FOR TEXT CLASSIFICATION

The predictive modeling problem of sequence classification involves the task of predicting a category for the given sequence where one has some sequences of inputs over space or time and the problem becomes difficult because sequences generally vary in length and can be formed of very large vocabulary of input symbols which may obligate the model to learn the dependencies between symbols in the sequence.

2.1 LSTM

In recurrent neural networks during the gradient back propagation phase, the gradient signal can be multiplied many times as the number of time steps by the weight matrix associated with the connections between the neurons of the recurrent hidden layer, in a traditional recurrent neural network. Thus, the magnitude of weights in the transition matrix may have a heavy impact on the learning process. If the leading eigenvalue of the weight matrix is smaller than 1.0, it may rise to a situation called 'vanishing gradients' where the gradient signal becomes so small that learning either become very slow or stops working completely. The task of learning long-term dependencies in the data is also be made more difficult. To the contrary, if the weights in this matrix are large or the eigenvalue of the weight matrix is larger than 1.0, it can lead to a situation where the gradient signal is so large that it can cause learning to diverge. This is often referred to as exploding gradients. These conditions are the main motivation behind the LSTM model where a new structure called a memory cell was introduced. The main components of a memory cell are - input gate, a neuron with a self-recurrent connection, forget gate, and output gate. The self-recurrent connection weighs 1.0 and verifies that, other than any outside interference, the state of a memory cell can remain constant from one time step to another. The gates modulate the interactions between the memory cell itself and its environment. The input gate allows incoming signal to alter the state of the memory cell or block it. On the other hand, the output gate allows the state of the memory cell to have an effect on other neurons or prevent it. Finally, the forget gate modulates the memory cell's self-recurrent connection, then allowing the cell to remember or forget its previous state, as required.

2.2 WORD EMBEDDING

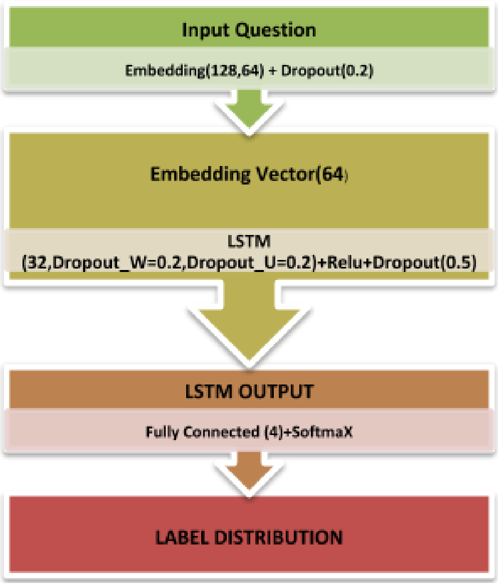
We will map each question into a real vector domain, a popular technique when working with text called word embedding. This is a technique where words are encoded as real-valued vectors in a high dimensional space, where the similarity between words in terms of meaning translates to closeness in the vector space. We will map each word onto a 64 length real valued vector. We will also limit the total number of words that we are interested in modelling to the 128 most frequent words, and zero out the rest. Since all the inputs are of variable lengths so we will limit the length to 500 words and truncate the larger inputs and zero pad the shorter inputs.
Now that we have defined our problem and how the data will be prepared and modelled, we are ready to develop an LSTM model to classify the SD classes of questions.

2.3 WORD2VEC EMBEDDING APPROACH

Word2vec model map each unique word id to a low-dimensional continuous vector-space based on their observed distributional properties in some text corpus. This vectors can be interpreted as points tracing out on the outside surface of a manifold in the embedded space.Word2vec is not a monolithic algorithm but rather involves two processes 'CBOW' and 'skip-gram'. Both of them map word(s) to target variables which is also word(s) and learn weights that acts as vector representation for the words.

III. ARCHITECTURE

The following archetype was used for both the embedding approaches. Since recurrent neural network and more generally LSTM suffers a lot from over fitting so we propose the use of dropout to regulate it. Note that we use dropout in the input and recurrent connections of the memory units with the LSTM precisely and separately and between the layers.


Input Question
Embedding(128,64) + Dropout(0.2)
Embedding Vector(64)
LSTM
(32,Dropout_W=0.2,Dropout_U=0.2)+Relu+Dropout(0.5)
LSTM OUTPUT
Fully Connected (4)+SoftmaX
LABEL DISTRIBUTION

IV. RESULTS

The table below displays the accuracy of different embedding approaches with the given architecture on the test dataset with train-test split of 80:20 and validation split of 0.1.

Embedding Approach	Accuracy
Word Embedding	78.3%
Word2Vec	80.4%

V. CONCLUSION AND FUTURE WORK

We see that our architecture with both the embedding approach show a decent score despite of the small dataset size. In our future work, we are focusing on character level representation of text and classification of it using a small Convolutional Neural Network.

VI. REFERENCES

[1]. I. Ikonamakis (2005), "Text Classification Using Machine Learning Techniques", WSEAS TRANSACTIONS on COMPUTERS, Vol.4, Issue.8, pp. 966-974.

[2].Ath. Kehagias, V. Petridis, V.G. Kaburlasos, and P. Fragkou, "A Comparison of Work and Sense-based Text Categorization Using Several Classification Algorithms" 6 April 2001.

[3].Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, "A neural probabilistic language model," The Journal of Machine Learning Research, vol. 3, pp. 1137–1155, 2003.

[4].Dong Wu and Mingmin Chi, "Long Short-Term Memory with Quadratic Connections in Recursive Neural Networks for representing Compositional Semantics ", Journal of IEEE Access, 201

[5].Dang Li. Jiang Qian School of Science, "Text Sentiment Analysis Based on Long Short-Term Memory", First IEEE International Conference on Computer Communication and the Internet, 2016.